

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Original) A method of performing transactions between disparate systems in a computing environment, said method comprising steps of:
 - (a) receiving a first message in a first message format from an originating system, said first message having request data indicative of a transaction request;
 - (b) at a first interface, evaluating said first message to ascertain said request data;
 - (c) said first interface further applying a predetermined criteria to said request data so as to generate said transaction request;
 - (d) routing said transaction request to an appropriate service system to be responsively fulfilled;
 - (e) fulfilling said transaction request at said service system and indicating same to said first interface; and

(f) issuing via said first interface a second message to said originating system in said first message format as a first response message to said transaction request.

2. (Currently amended) The A method as set forth in claim 1, wherein said first interface includes a network agent and an interface agent, said network agent being operative to extract said request data from said first message and said interface agent being operative to produce said transaction request therefrom.

3. (Currently amended) The A method as set forth in claim 2, wherein said service system includes a server agent in communication with said interface agent.

4. (Currently amended) The A method as set forth in claim 3, wherein said agents communicate with each other via message queues.

5. (Currently amended) The A method as set forth in claim 1, wherein said first interface functions to generate a context data structure organizationally independent of said first message format, said context data structure being acted upon in servicing said transaction request.

6. (Currently amended) The A method as set forth in claim 5, wherein said transaction request is routed to said service system through application of topological configuration objects to a predetermined ruleset.

7. (Currently amended) The A method as set forth in claim 6, wherein said ~~predefined~~ topological configuration objects are modifiable independently of said ruleset.

8. (Currently amended) The A method as set forth in claim 1, wherein said step of fulfilling said transaction request comprises:

(g) at a second interface, issuing a third message in a second message format to an auxiliary system requesting information;

(h) receiving a second response message at said second interface in said second message format containing said information; and

(i) extracting said information from said second message at said second interface and providing said information to said service system.

9. (Currently amended) The A method as set forth in claim 8, wherein said first message format and said second message format are different protocols.

10. (Currently amended) The A method as set forth in claim 8, wherein said first message format and said second message format are equivalent protocols.

11. (Currently amended) In a computing environment including a plurality of computer systems, a system an arrangement comprising:

each of said computer systems being equipped with agents operative to transfer external messages into and out of a particular message format;

each of said computer systems further having stored configuration objects indicating expected transaction requests and corresponding service systems; and

such that transaction requests are disassociated from incoming message by said agents and configuration objects, said transaction requests being automatically routed to appropriate service systems for fulfillment.

12. (Currently amended) The system An arrangement as set forth in claim 11, wherein said agents include:

network agents operative to extract request data from said external messages;

interface agents operative to produce said transaction request from said request data; and

server agents in communication with said interface agents so as to fulfill said transaction requests.

13. (Currently amended) The system ~~An arrangement~~ as set forth in claim 12, wherein said agents communicate with each other using message queues.

14. (Currently amended) The system ~~An arrangement~~ as set forth in claim 11, wherein said agents function to generate a context data structure organizationally independent of said external messages, said context data structure being acted upon in fulfilling said transaction request.

15. (Currently amended) The system ~~An arrangement~~ as set forth in claim 11, wherein said configuration objects are easily modifiable to reflect changes in said computing environment.

16. (Currently amended) A system ~~An arrangement~~ for performing transactions between a first computer entity and a second computer entity having disparate operating systems, said arrangement comprising:

a first interface running first agents according to predetermined configuration objects;

a second interface running second agents according to said predetermined configuration objects;

said first interface being operative to translate a first message in a first message format received from said first computer entity into an organizationally independent context data structure; and

said second interface reading and acting upon said context data structure to produce a second message in a second message format issued to said second computer entity.

17. (Currently amended) The system ~~An arrangement~~ as set forth in claim 16, wherein:

said second computer entity issues a third message in said second message format in response to said second message, said second interface being operative to update said context data structure based thereon; and

said first interface responsively acting to read said context data structure and produce a fourth message in said first message format back to said first computer entity.

18. (Currently amended) The system ~~An arrangement~~ as set forth in claim 16, wherein said context data structure includes a hash table containing pointers to a plurality of element value locations.